

Ex 3 Data Visualization through Python

September 4, 2023

Exp.No: 3

URK21CS1128

DATA VISUALIZATION THROUGH PYTHON

Aim: To execute the basic functionalities using data visualization with various charts.

Description:

Data visualization provides a good, organized pictorial representation of the data which makes it easier to understand, observe, analyze. In this tutorial, we will discuss how to visualize data using Python. Python provides various libraries that come with different features for visualizing data. All these libraries come with different features and can support various types of graphs. In this tutorial, we will be discussing four such libraries.

Matplotlib: Matplotlib is an easy-to-use, low-level data visualization library that is built on NumPy arrays. It consists of various plots like scatter plot, line plot, histogram, etc. Matplotlib provides a lot of flexibility.

Scatter Plot: Scatter plots are used to observe relationships between variables and uses dots to represent the relationship between them. The `scatter()` method in the matplotlib library is used to draw a scatter plot.

Line Chart: Line Chart is used to represent a relationship between two data X and Y on a different axis. It is plotted using the `plot()` function.

Bar Chart: A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. It can be created using the `bar()` method.

Histogram: A histogram is basically used to represent data in the form of some groups. It is a type of bar plot where the X-axis represents the bin ranges while the Y-axis gives information about frequency. The `hist()` function is used to compute and create a histogram.

Program:

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_csv("Emp_visu.csv")
print("URK21CS1128")
df
```

URK21CS1128

```
[1]:      First Name  Gender  Salary  Bonus %  Senior Management  \
0        Maria  Female  130590   11.858             False
1        Angela  Female   54568   18.523             True
2         Allan   Male  125792    5.042             False
3         Rohan  Female   45906   11.598             True
4       Douglas   Male   97308    6.945             True
5       Brandon   Male  112807   17.492             True
6         Diana  Female  132940   19.082             False
7       Frances  Female  139852    7.524             True
8       Matthew   Male  100612   13.645             False
9         Larry   Male  101004    1.389             True
10      Joshua   Male   90816   18.816             True
11       Jerry   Male   72000    9.340             True
12       Lois   Female   64714    4.934             True
13      Dennis   Male  115163   10.125             False
14       John    Male   97950   13.873             False
15      Thomas   Male   61933   10.945             True
16      Shawn    Male  111737    6.414             False
17       Gary    Male  109831    5.831             False
18      Jeremy   Male   90370    7.369             False
19  Kimberly   Female   41426    7.450             True
20      Louise   Female   63241   15.132             True
21      Donna   Female   81014    1.894             False
22       Ruby   Female   65476   10.012             True
23    Lillian   Female   59414    1.256             False
24      Julie   Female  102508   12.637             True
```

	Team	Age	Experience	New_Salary	Incentive
0	Finance	26	5	146075.36220	20000
1	Business Development	27	5	64675.63064	19000
2	Client Services	28	6	132134.43260	18500
3	Finance	28	7	51230.17788	18000
4	Marketing	28	7	104066.04060	17000
5	Human Resources	30	8	132539.20040	16000
6	Client Services	31	9	158307.61080	15800
7	Business Development	34	10	150374.46450	15500
8	Marketing	34	10	114340.50740	15000
9	Client Services	35	11	102406.94560	14700
10	Client Services	35	11	107903.93860	14300
11	Finance	35	12	78724.80000	14000
12	Legal	35	12	67906.98876	14000
13	Legal	36	13	126823.25380	13000
14	Client Services	37	13	111538.60350	12000
15	Marketing	38	14	68711.56685	11900
16	Human Resources	39	15	118903.81120	11500

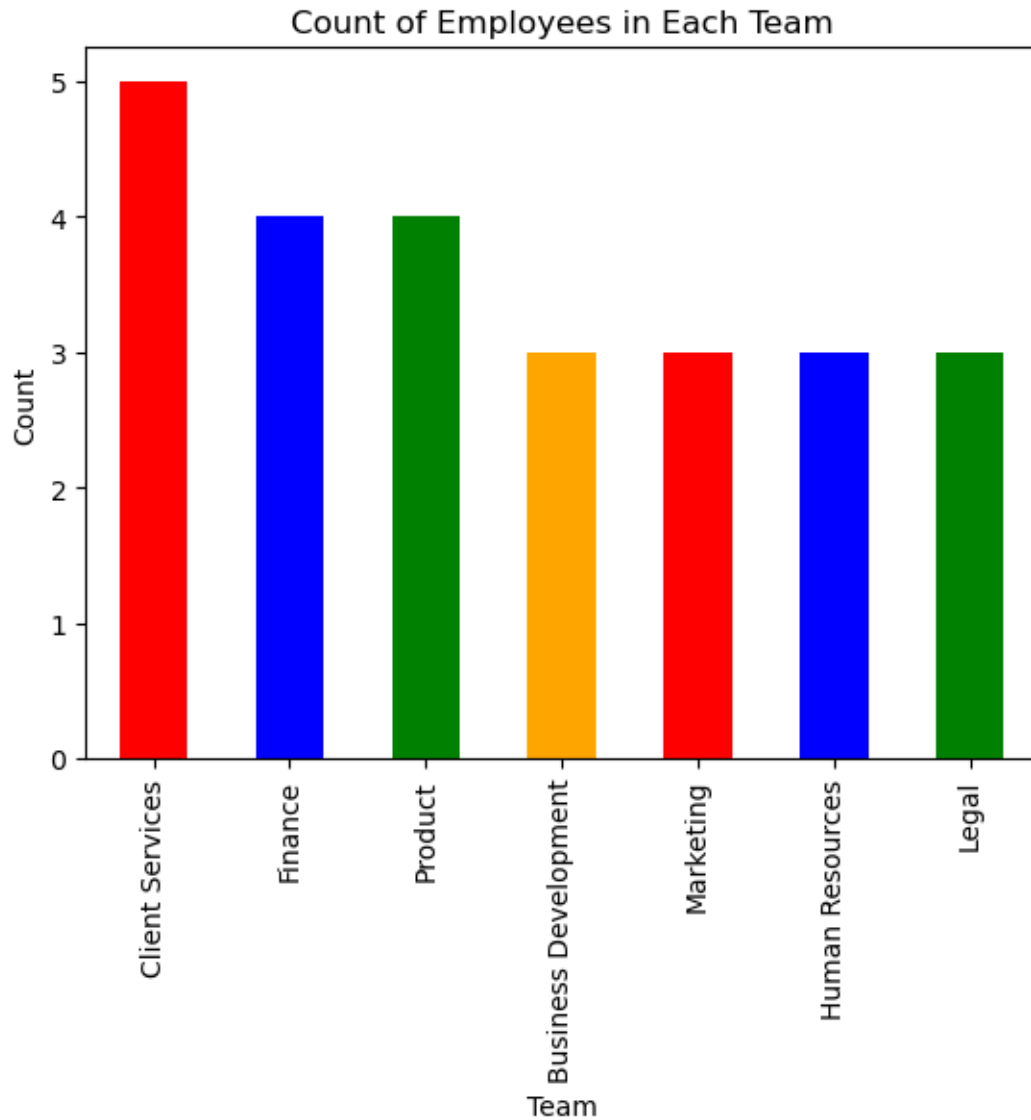
17	Product	39	15	116235.24560	11500
18	Human Resources	42	18	97029.36530	11000
19	Finance	44	20	44512.23700	11000
20	Business Development	45	21	72810.62812	10800
21	Product	49	23	82548.40516	10600
22	Product	54	25	72031.45712	10400
23	Product	55	26	60160.23984	10300
24	Legal	58	27	115461.93600	10000

Q1: Draw a bar chart with Team and its count (use different colors for each team)

```
[4]: print("URK21CS1128")
team_count = df['Team'].value_counts()
team_count.plot(kind='bar', color=['red', 'blue', 'green', 'orange'])
plt.xlabel('Team')
plt.ylabel('Count')
plt.title('Count of Employees in Each Team')
```

URK21CS1128

```
[4]: Text(0.5, 1.0, 'Count of Employees in Each Team')
```

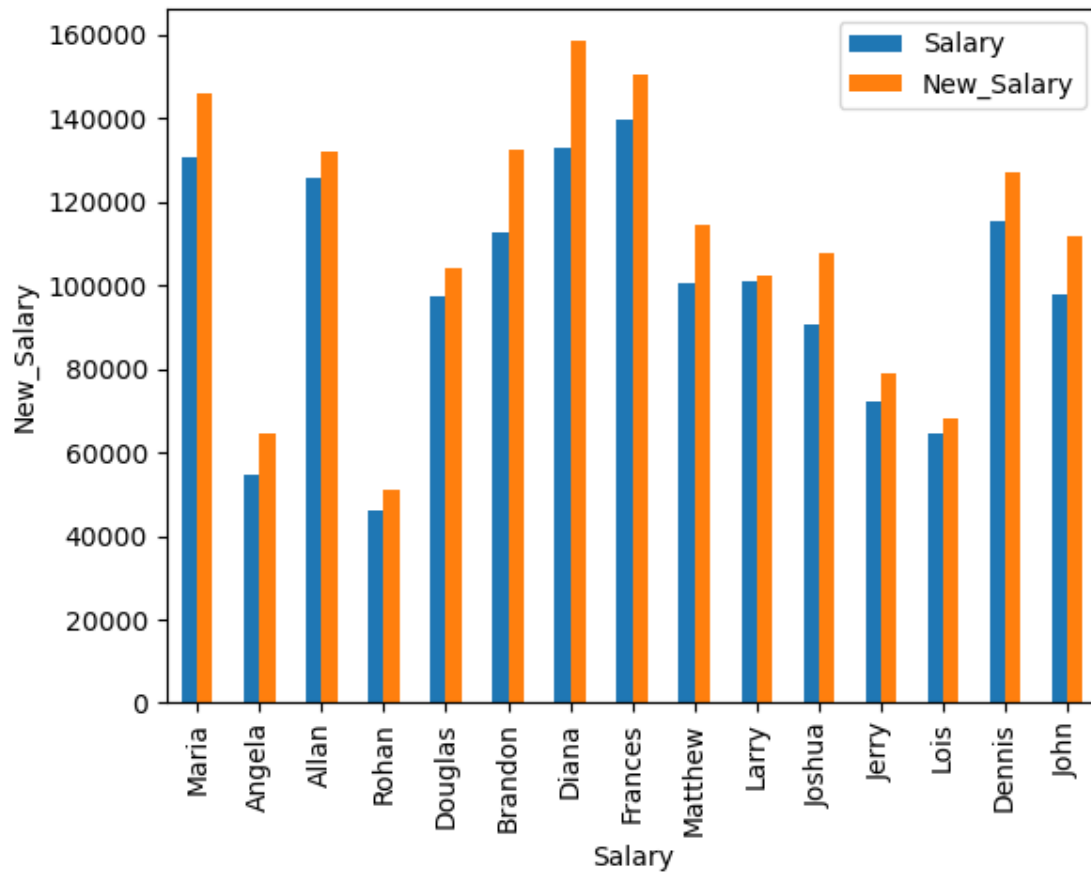


Q2: Draw a comparative bar chart for Salary and New_Salary against each person (first 15 persons)

```
[5]: print("URK21CS1128")
df.head(15).plot(x='First Name', y=['Salary', 'New_Salary'], kind='bar')
plt.xlabel('Salary')
plt.ylabel('New_Salary')
```

URK21CS1128

```
[5]: Text(0, 0.5, 'New_Salary')
```

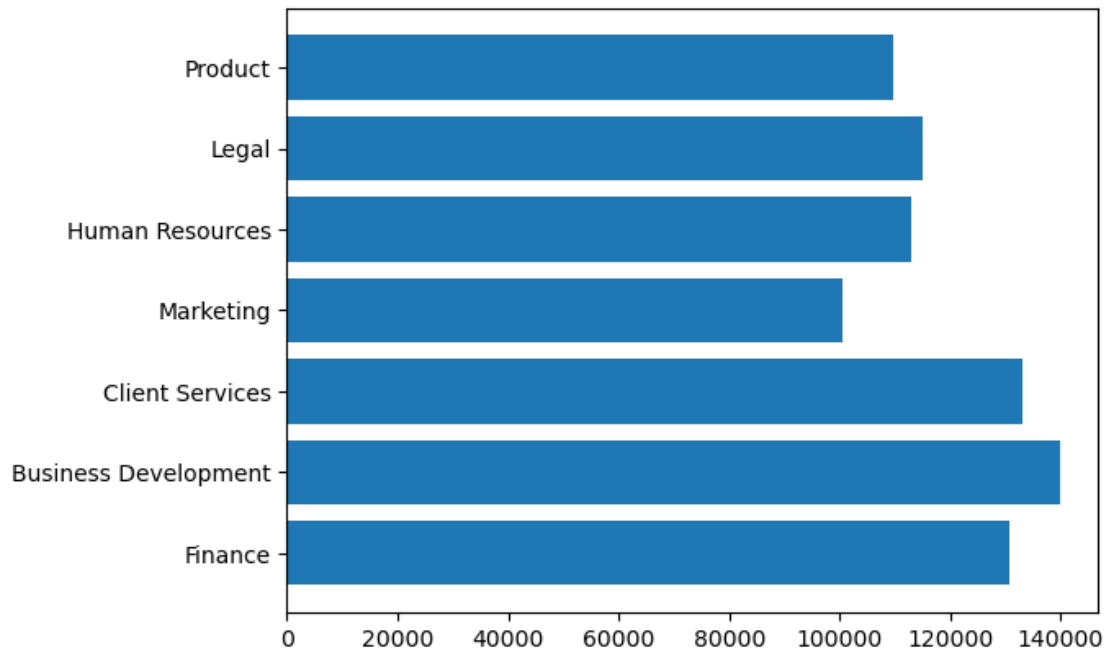


Q3: Draw a horizontal bar chart for Team and Salary

```
[6]: print("URK21CS1128")
plt.barh(df["Team"],df["Salary"])
```

URK21CS1128

[6]: <BarContainer object of 25 artists>

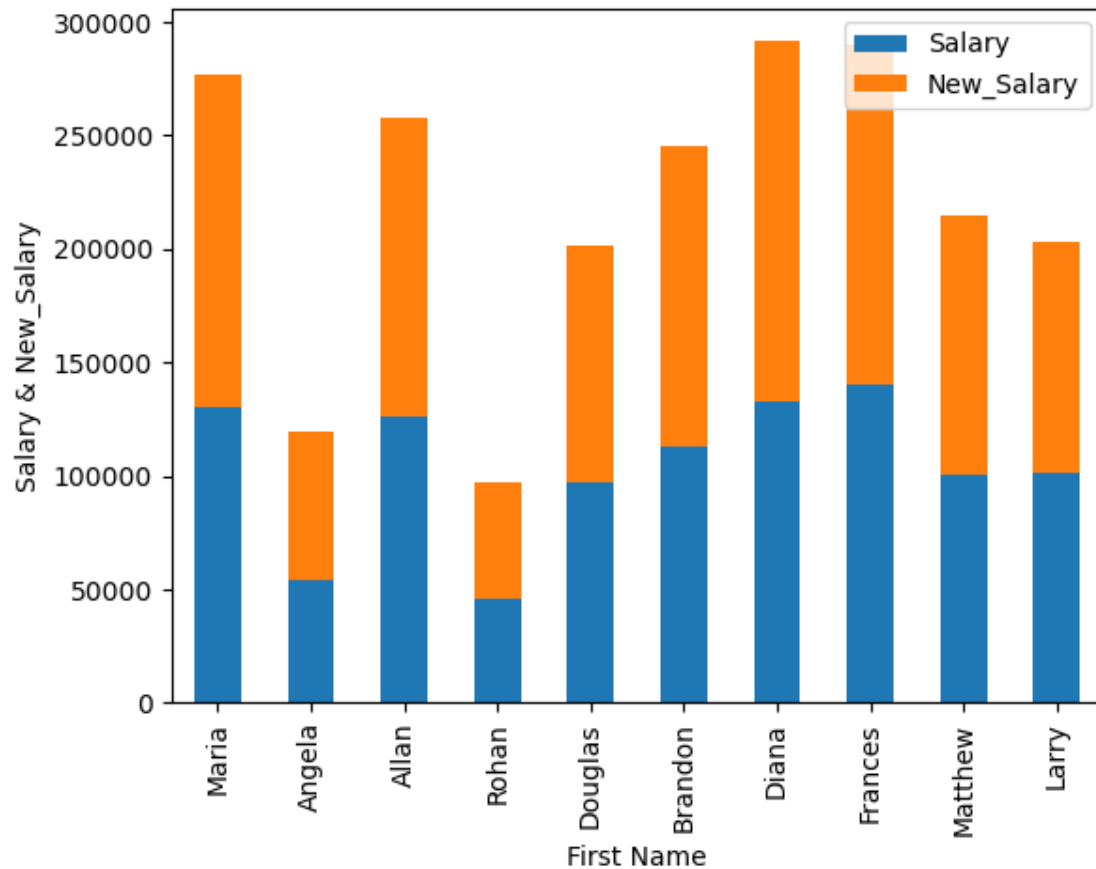


Q4: Draw a stacked bar chart for Salary and New_price against the person (first 10 persons)

```
[7]: print("URK21CS1128")
df.head(10).plot(x='First Name', y=['Salary', 'New_Salary'], kind='bar',
    ↪stacked=True)
plt.xlabel('First Name')
plt.ylabel('Salary & New_Salary')
plt.show()

# plt.figure(figsize = (8,5))
# plt.bar(df['First Name'].head(10),df['Salary'].head(10))
# plt.bar(df['First Name'].head(10),df['New_Salary'].
    ↪head(10),bottom=df['Salary'].head(10))
```

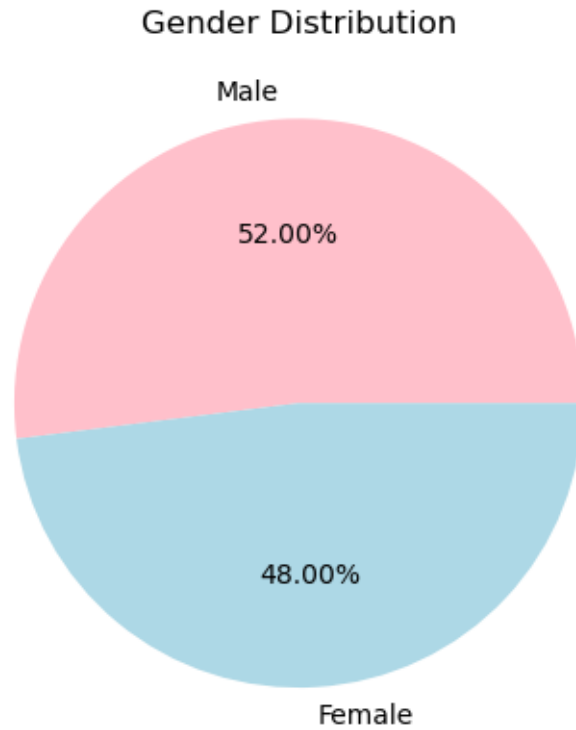
URK21CS1128



Q5: Draw a pie chart with Gender and its count

```
[8]: print("URK21CS1128")
a = df['Gender'].value_counts()
plt.pie(df['Gender'].value_counts(), labels=a.index, autopct='%1.2f%%',
        colors=['pink', 'lightblue'])
plt.title('Gender Distribution')
plt.show()
```

URK21CS1128

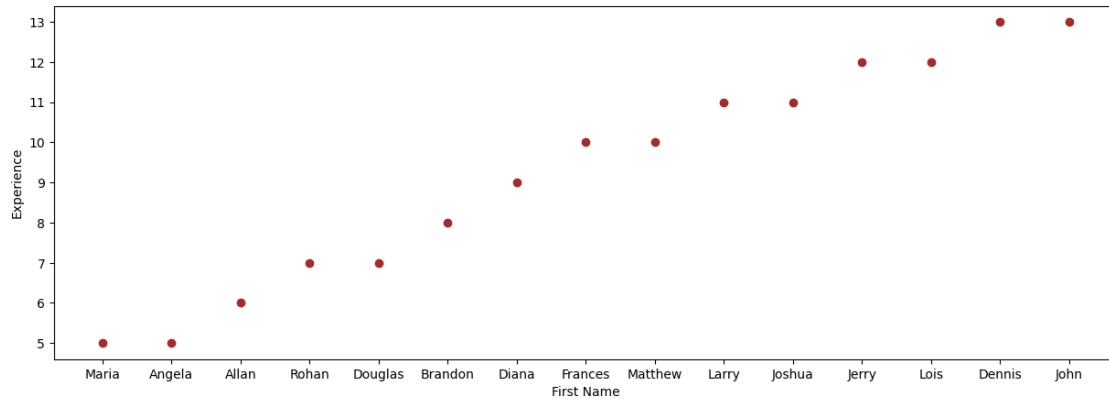


Q6: Draw the dot plot between person and experience (first 15 persons)

```
[9]: print("URK21CS1128")
plt.figure(figsize = (15,5))
plt.plot(df["First Name"].head(15),df["Experience"].head(15),color =_
↪"brown",marker = 'o',linewidth=0)
plt.xlabel("First Name")
plt.ylabel("Experience")
```

URK21CS1128

```
[9]: Text(0, 0.5, 'Experience')
```

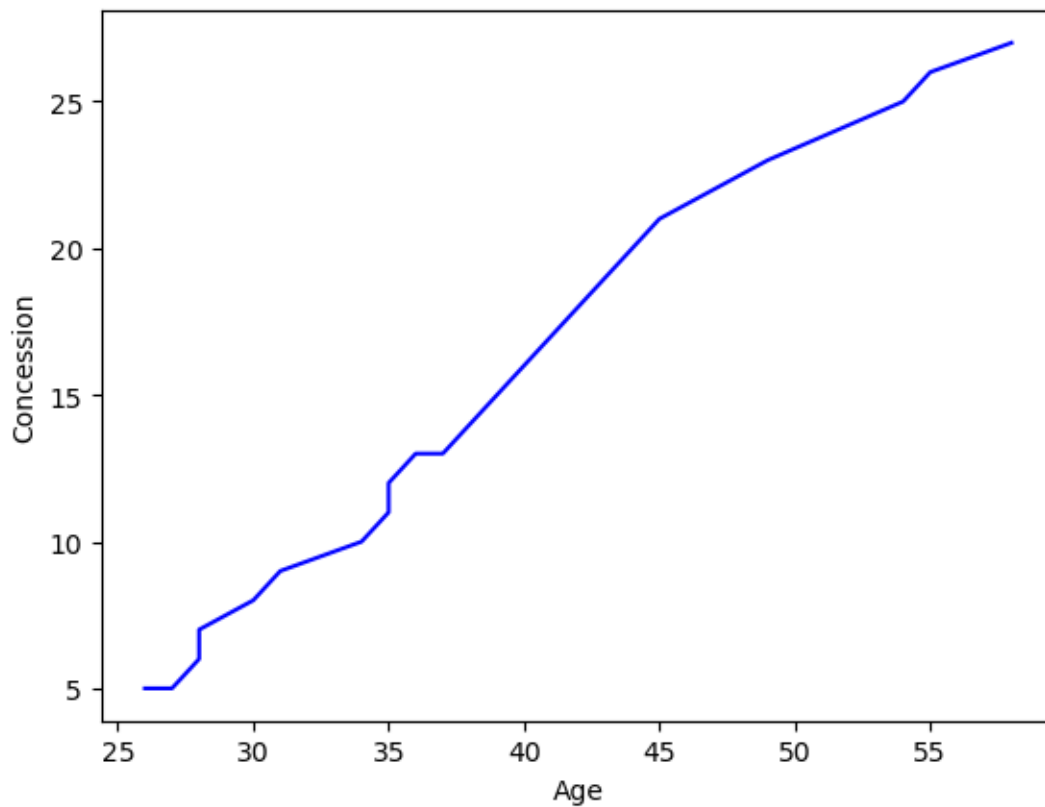



Q7: Draw the line plot between age and experience. Observe the trend line.

```
[10]: print("URK21CS1128")
plt.plot(df["Age"],df["Experience"],color = "blue")
plt.xlabel("Age")
plt.ylabel("Concession")
```

URK21CS1128

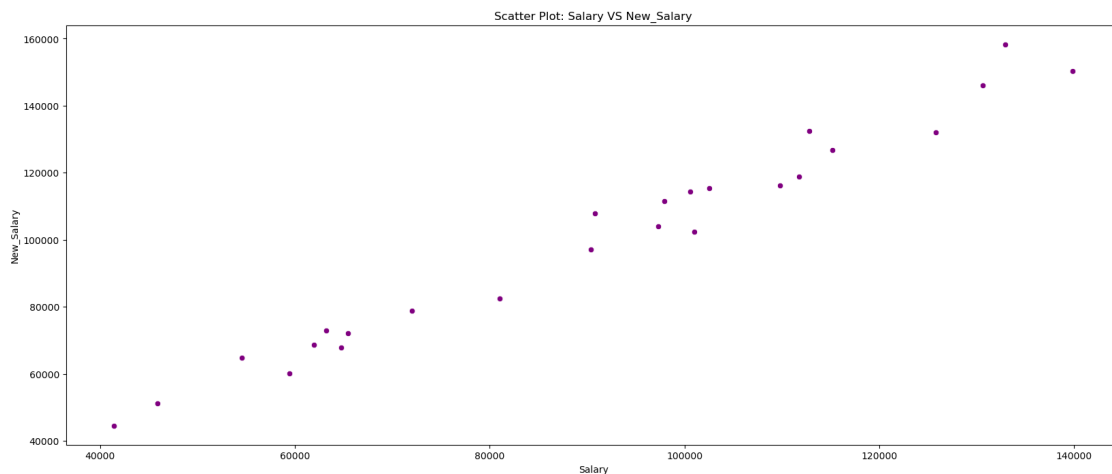
```
[10]: Text(0, 0.5, 'Concession')
```



Q8: Draw the scatter plot between Salary and New_Salary. Observe the correlation

```
[11]: print("URK21CS1128")
import seaborn as sns
plt.figure(figsize=(20,8))
sns.scatterplot(df, x='Salary', y='New_Salary', color='Purple')
plt.title('Scatter Plot: Salary VS New_Salary')
plt.xlabel('Salary')
plt.ylabel('New_Salary')
plt.show()
```

URK21CS1128

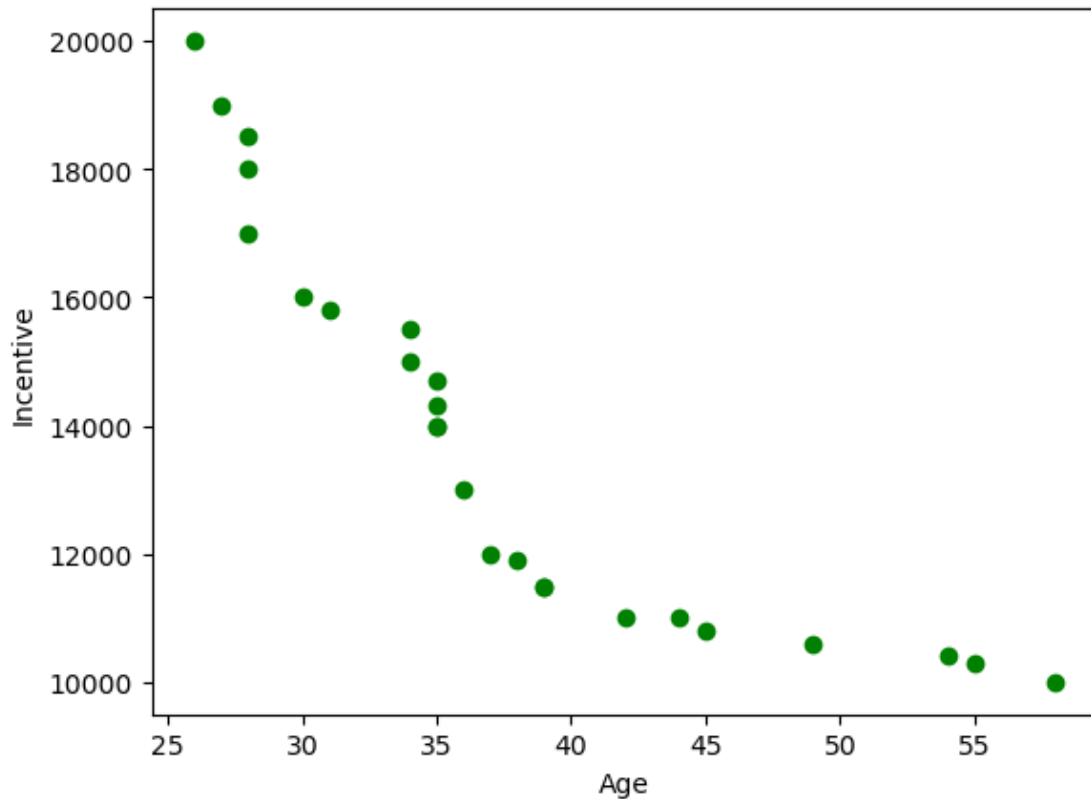


Q9: Draw the scatter plot between Age and Incentive. Observe the correlation

```
[12]: print("URK21CS1128")
plt.scatter(df["Age"],df["Incentive"],color='green')
plt.xlabel("Age")
plt.ylabel("Incentive")
```

URK21CS1128

```
[12]: Text(0, 0.5, 'Incentive')
```



Q10: Draw the box plot to show the statistical summary of Age column

```
[13]: print("URK21CS1128")
      df["Age"].plot.box()
      df.describe()
```

URK21CS1128

```
[13]:
```

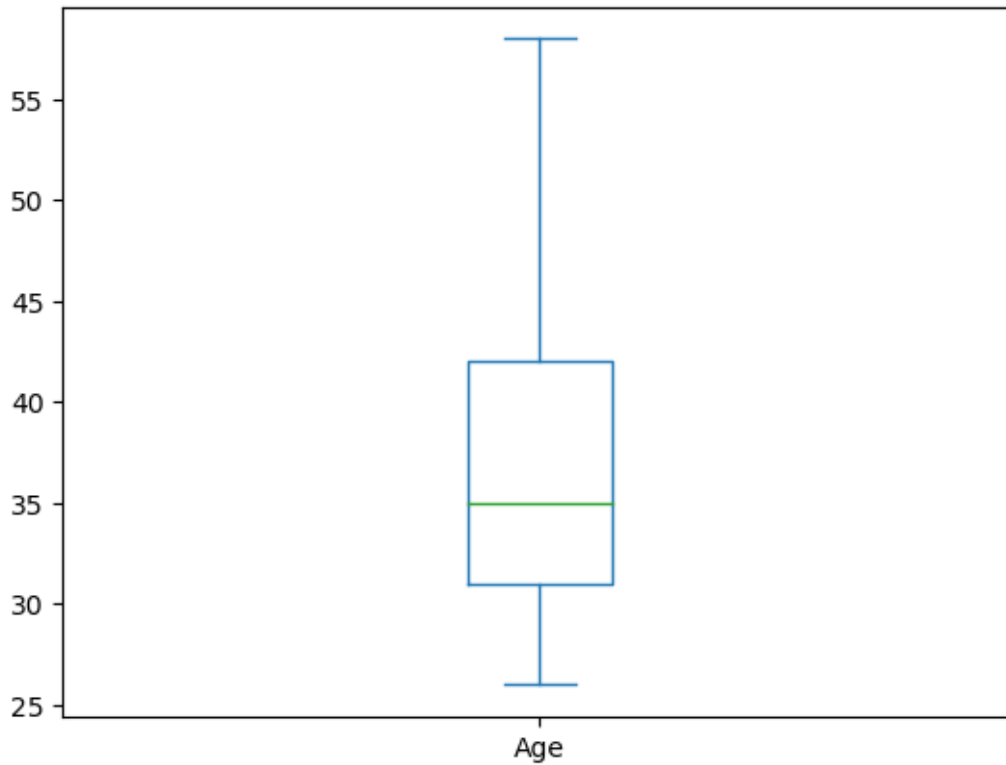
	Salary	Bonus %	Age	Experience	New_Salary \
count	25.000000	25.000000	25.000000	25.00000	25.000000
mean	90758.880000	9.965040	37.680000	13.72000	99898.113979
std	28441.424571	5.336828	8.938307	6.64906	32108.798871
min	41426.000000	1.256000	26.000000	5.00000	44512.237000
25%	64714.000000	6.414000	31.000000	9.00000	72031.457120
50%	97308.000000	10.012000	35.000000	12.00000	104066.040600
75%	111737.000000	13.645000	42.000000	18.00000	118903.811200
max	139852.000000	19.082000	58.000000	27.00000	158307.610800

	Incentive
count	25.000000
mean	13832.000000
std	3034.347816

```

min    10000.000000
25%    11000.000000
50%    14000.000000
75%    15800.000000
max     20000.000000

```



Q11: Draw the histogram plot for Experience column

```

[14]: print("URK21CS1128")
      plt.hist(df["Experience"], edgecolor='white')

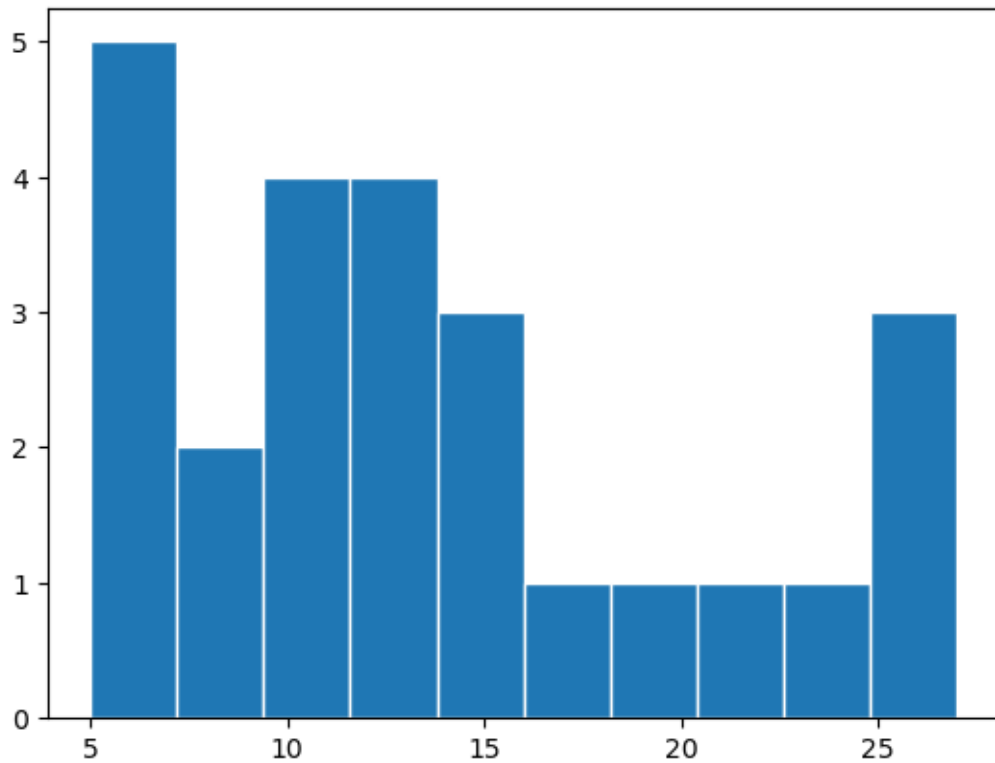
```

URK21CS1128

```

[14]: (array([5., 2., 4., 4., 3., 1., 1., 1., 1., 3.]),
      array([ 5. ,  7.2,  9.4, 11.6, 13.8, 16. , 18.2, 20.4, 22.6, 24.8, 27. ])),
      <BarContainer object of 10 artists>)

```

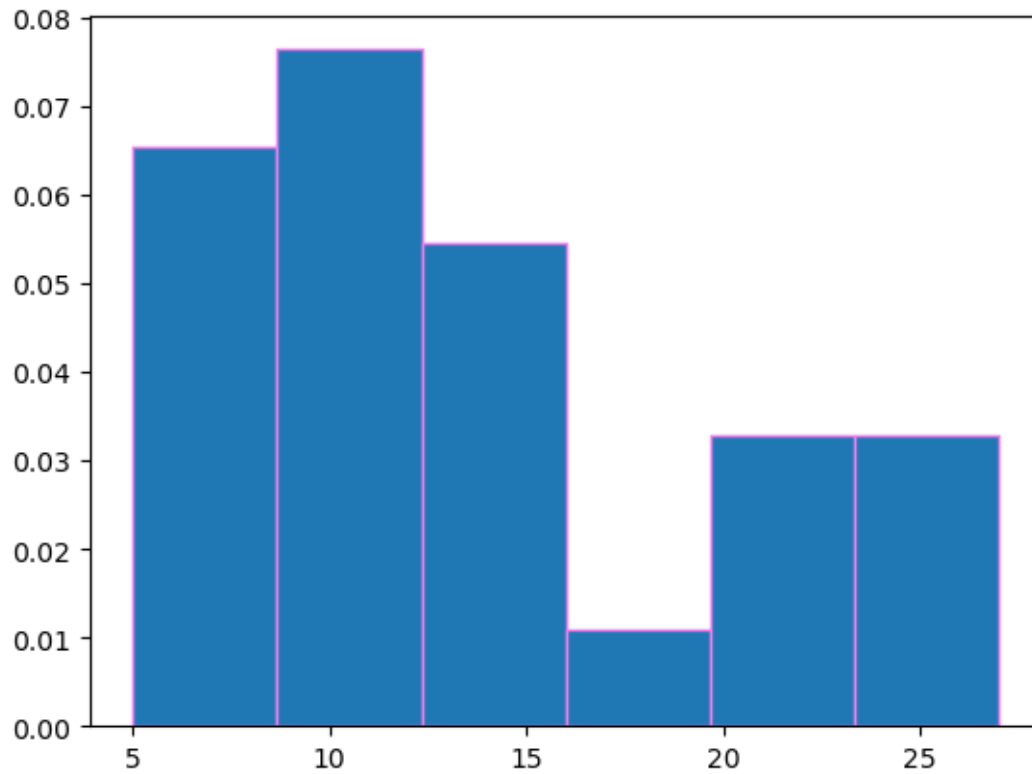


Q12: Draw the histogram plot for Experience column with bin value and PDF

```
[15]: print("URK21CS1128")
plt.hist(df["Experience"],density='True', bins=6, edgecolor='violet')
```

URK21CS1128

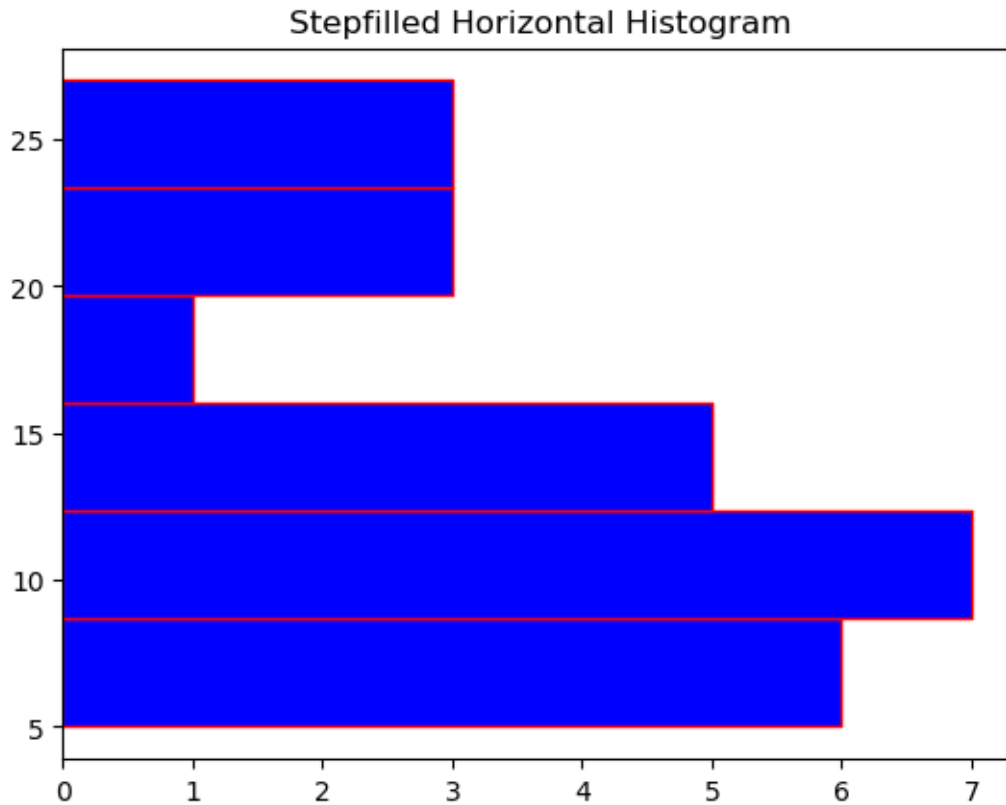
```
[15]: (array([0.06545455, 0.07636364, 0.05454545, 0.01090909, 0.03272727,
0.03272727]),
array([ 5.          ,  8.66666667, 12.33333333, 16.          , 19.66666667,
23.33333333, 27.          ]),
<BarContainer object of 6 artists>)
```



13. Write code to change the horizontal histogram

```
[16]: print("URK21CS1128")
plt.hist(df['Experience'], bins=6, orientation='horizontal', color='blue', edgecolor='red')
plt.title('Stepfilled Horizontal Histogram')
plt.show()
```

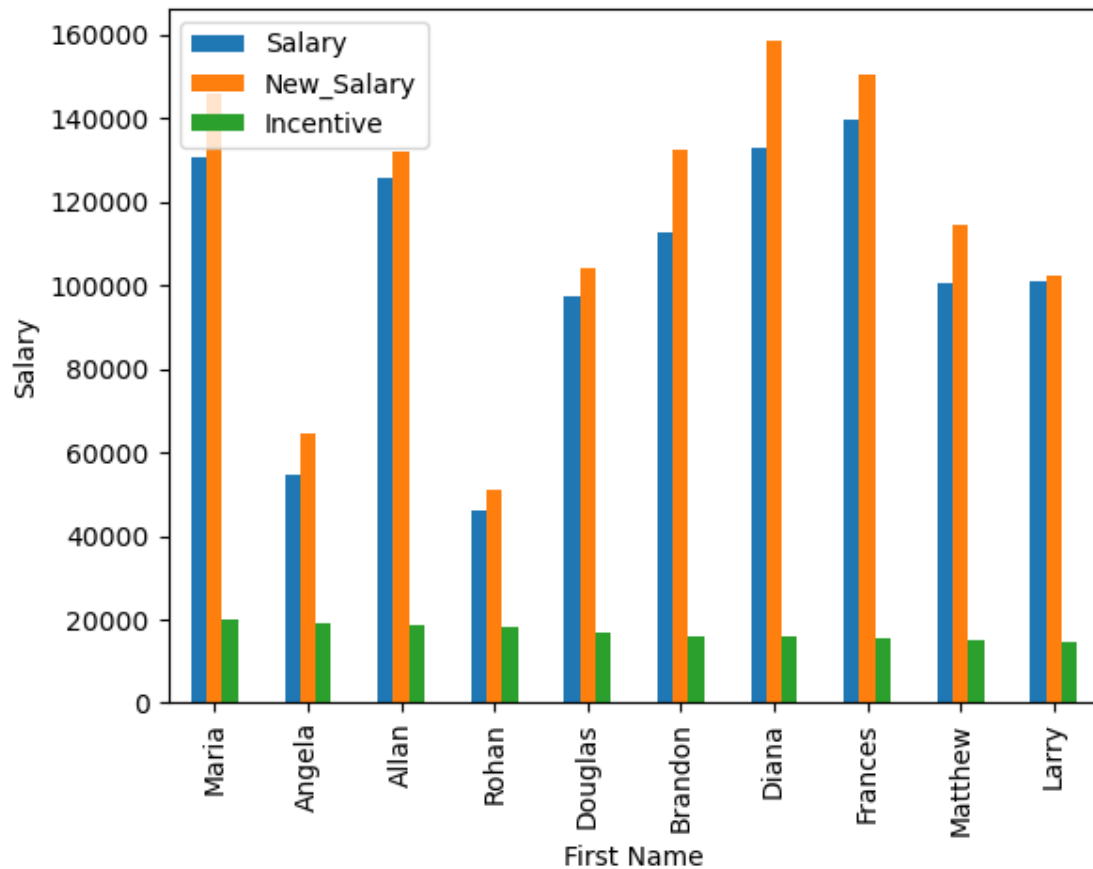
URK21CS1128



14. Compare any three features and display the comparative bar graph

```
[17]: print("URK21CS1128")
comparison_data = df.head(10)[['First Name', 'Salary', 'New_Salary', 'Incentive']]
comparison_data.set_index('First Name').plot(kind='bar')
plt.xlabel('First Name')
plt.ylabel('Salary')
plt.show()
```

URK21CS1128

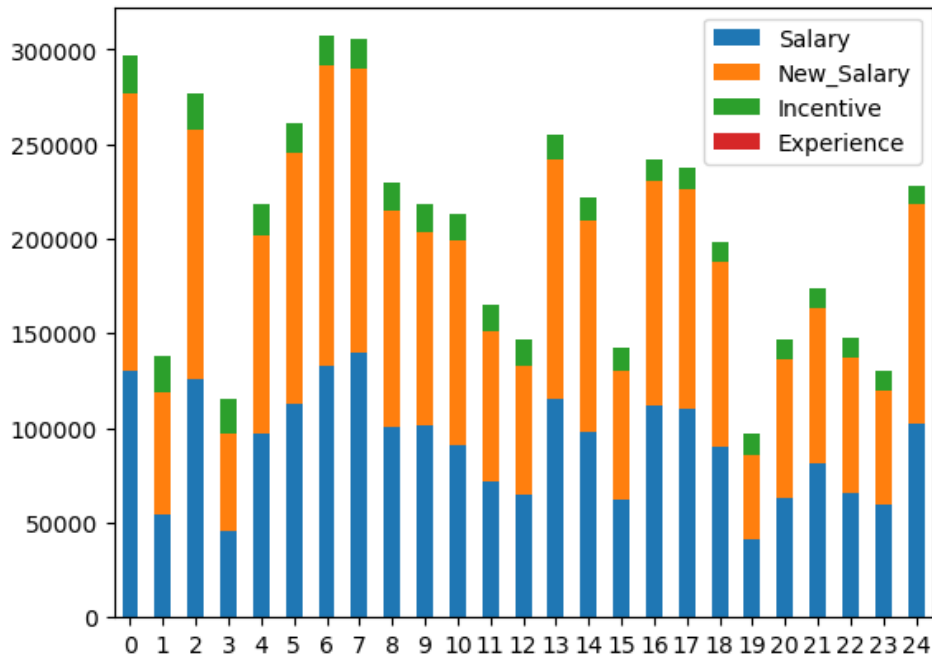


15. Stack any 4 features using a bar chart

```
[18]: print("URK21CS1128")
df[['Salary', 'New_Salary', 'Incentive', 'Experience']].plot(kind='bar',
    ↪stacked=True)
plt.title('Stacked Bar Chart: Distribution of Salary, New Salary, Incentive,
    ↪and New Price')
plt.xticks(rotation=0)
plt.legend()
plt.show()
```

URK21CS1128

Stacked Bar Chart: Distribution of Salary, New Salary, Incentive, and New Price



Result: The basic functionalities of data visualization using python were executed successfully.